What Your Images Reveal: Exploiting Visual Contents for Point-of-Interest Recommendation

Suhang Wang[†], Yilin Wang[†], Jiliang Tang[‡], Kai Shu[†], Suhas Ranganath[†], Huan Liu[†]
[†]Computer Science & Engineering, Arizona State University, Tempe, AZ, USA
[‡]Computer Science & Engineering, Michigan State University, East Lansing, MI, USA
[†]{suhang.wang, yilin.wang, kai.shu, suhas.ranganath, huan.liu}@asu.edu,
[‡]tangjili@msu.edu

ABSTRACT

The rapid growth of Location-based Social Networks (LBSNs) provides a vast amount of check-in data, which facilitates the study of point-of-interest (POI) recommendation. The majority of the existing POI recommendation methods focus on four aspects, i.e., temporal patterns, geographical influence, social correlations and textual content indications. For example, user's visits to locations have temporal patterns and users are likely to visit POIs near them. In real-world LBSNs such as Instagram, users can upload photos associating with locations. Photos not only reflect users' interests but also provide informative descriptions about locations. For example, a user who posts many architecture photos is more likely to visit famous landmarks; while a user posts lots of images about food has more incentive to visit restaurants. Thus, images have potentials to improve the performance of POI recommendation. However, little work exists for POI recommendation by exploiting images. In this paper, we study the problem of enhancing POI recommendation with visual contents. In particular, we propose a new framework Visual Content Enhanced POI recommendation (VPOI), which incorporates visual contents for POI recommendations. Experimental results on real-world datasets demonstrate the effectiveness of the proposed framework.

Keywords

POI recommendation; Visual contents; Location-based Social Networks

1. INTRODUCTION

As an increasingly popular application of location-based services, location-based social networks (LBSNs), such as Yelp, Instagram and Foursquare, have attracted millions of users. Users in LBSNs can check in their preferred points-of-interest (POIs), e.g., museums, restaurants and stores, and share their experiences of visiting these POIs with friends, resulting in huge amount of user check-in data. The availability of user check-in data in large volume brings in new opportunities to design appealing services to facilitate user's travels and social interactions. Personalized POI recommendation, which aims at recommending personalized POIs to a user who has not visited them before, is one of such services. Various POI recommendation methods have been proposed, which mainly study four aspects, i.e., geographical influence, social correlations, temporal patterns and textual content indications [8, 37, 35, 4, 34, 10]. These aspects have been proven to be effective for improving POI recommendations.

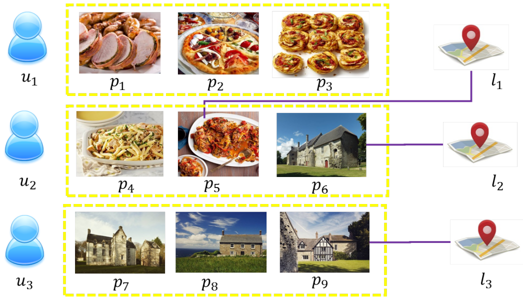


Figure 1: An Example of Images Posted by Users.

As the growth of LBSNs, more and more LBSNs are now encouraging users to associate POIs with images. For example, Yelp users can check-in a POI and *add images* to that check-in. When uploading images to Instagram, users can choose to *add locations* to images. Images associated with POIs contain rich unique information about user preferences and POI properties, such as shapes, structures and textures of POIs that are not available in the aforementioned four aspects. Thus, images can provide added value to improve the performance of POI recommendations. Figure 1 gives an example of images posted by users in Instagram, where images in the three yellow dashed rectangles are posted by users u_1, u_2, and u_3, respectively. The purple line connecting an image and a location means that the image is associated/tagged with the location. In the figure, u_3 posts lots of architectures, which implies that u_3 likes architectures of the style shown in images. Location l_2 is associated with p_6, and p_6 contains visual contents, such as shapes, structures and textures that are similar to images posted by u_3, which suggests that l_2 could satisfy u_3's interests and we may recommend l_2 to u_3. u_1 is a cold-start user without any check-in records. Based on the food images posted by u_1, we still could recommend l_1 to u_1 as the visual contents of images associated with l_1 indicate l_1 as a restaurant. Pre-

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052638

vious work also suggests that images and POIs have strong connections [16]. Thus, incorporating images could improve the performance of POI recommendations. However, there's little existing work on exploiting images for POI recommendations.

In this paper, we study the problem of enhancing POI recommendation with visual contents. In essence, we solve two challenges - (1) how to extract useful visual contents from images as we are lack of ground truth of what are contained in the images; and (2) how to incorporate visual contents for POI recommendations. In an attempt to solve these two challenges, we propose a novel POI recommendation framework called Visual Content Enhanced POI recommendation (VPOI). The major contributions of this paper are summarized next:

- Studying the new problem of enhancing POI recommendation using visual contents;
- Proposing a novel POI recommender system, which incorporates visual contents into a probabilistic model for learning user and POI latent features; and
- Conducting experiments on real-world datasets to demonstrate the effectiveness of the proposed framework.

The rest of the paper is organized as follows. In Section 2, we present related work. In Section 3, we introduce the proposed framework VPOI. In Section 4, we present a method to solve the optimization problem of VPOI. In Section 5, we show empirical evaluation with discussions. In Section 6, we give conclusion with future work.

2. RELATED WORK

In this section, we will briefly review related works on POI recommendation and visual contents for data mining.

2.1 POI Recommendation

POI recommendation, also called location recommendation, has been recognized as an essential task on recommender systems. Existing work on POI recommendation generally focuses on four aspects, i.e., geographical influence, social correlations, temporal patterns and textual content indications [8]. Ye et al. [36] introduced POI recommendation on LBSNs and investigated the geographical influence [37] and social influence [35] for POI recommendation. Cheng et al. [4] investigated the geographical and social influence through a multi-center Gaussian model. Zhang et al. [39] further exploits categorical correlations together with geographical and social correlations. Temporal information has also attracted much attention from researchers. Gao et al. [9] investigated the temporal cyclic patterns of check-ins in terms of temporal non-uniformness and temporal consecutiveness. Yuan et al. [38] incorporated both temporal cyclic information and geographical information for time-aware POI recommendation. Cheng et al. [5] introduced the task of successive personalized POI recommendation in LBSNs with a matrix factorization method. Recently, researchers started to explore the textual content information on LBSNs for POI recommendation. Yang et al. [34] introduced sentiment information and reported its better performance over state-of-the-art approaches. Liu et al. [18] studied the effect of POI-associated tags with an aggregated LDA model. Gao et al. [10] studied document content information on LBSNs w.r.t. POI properties, user interests, and sentiment indications. Though various aspects are investigated for POI recommendation, image contents haven't been studied for POI recommendation while image contents, i.e., visual contents, have been proven to be effective for many data mining tasks, which will be introduced next.

2.2 Visual Contents for Data Mining

The famous saying that "A picture is worth a thousand words" suggests that images posted on LBSNs contain rich information, which have potentials to facilitate data mining tasks such as recommendations. Recently, researchers have started to pay attention on investigating images for data mining tasks. McAuley et al. [21] developed a system that can recommend which clothes and accessories will go well together by utilizing visual contents extracted from cloth and accessory images. He et al. [13] studied evolving visual factors that people consider when evaluating products so as to make better recommendations. Wang et al. [31] infers sentiments from visual contents. In [32, 33], tags and labels of images are predicted from visual contents. The work on images and locations can be generally categorized into two types. One type is dividing the concerned region into grids and predicting the grid where an image resides [11, 19]. Hay and Efros [11] proposed a data driven approach to calibrate Flickr images to grids on the earth by using visual contents. The other type is associating images with landmarks or POIs [16, 6]. Li et. al [16] predicted which POI a photo is taken by using bag-of-words visual contents. The work on associating images with POIs suggests that images have strong connections with POIs and can be used to describe the properties of POIs.

Our problem is different from the aforementioned approaches. Instead of associating images with POIs, we use images to help learn the latent features of both users and POIs for the task of personalized POI recommendation.

3. A VISUAL CONTENT ENHANCED POI RECOMMENDER SYSTEM

Before introducing details about the proposed framework, we will first introduce notations used in this paper. Throughout this paper, matrices are written as boldface capital letters and vectors are denoted as boldface lowercase letters. For an arbitrary matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$, $\mathbf{M}_{ij}$ denotes the (i, j)-th entry of $\mathbf{M}$ while $\mathbf{m}_i$ and $\mathbf{m}^j$ mean the i-th column and j-th row of $\mathbf{M}$, respectively. $\|\mathbf{M}\|_F$ is the Frobenius norm of $\mathbf{M}$. Capital letters in calligraphic math font such as $\mathcal{P}$ are used to denote sets and $|\mathcal{P}|$ denotes the cardinality of $\mathcal{P}$.

There are three types of objects in the studied problem, namely, users, locations and images. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of users, $\mathcal{L} = \{l_1, l_2, \ldots, l_m\}$ be the set of locations and $\mathcal{P} = \{p_1, \ldots, p_N\}$ be the set of photos, where n, m and N are the number of users, POIs and images, respectively. Users can check in at locations. We use $\mathbf{X} \in \mathbb{R}^{n \times m}$ to denote user-POI check-in matrix. $\mathbf{X}_{ij}$ means the check-in frequency or rating of u_i on l_j. Following the common way to deal with check-in frequency [9, 10], we use $\mathbf{R} \in \mathbb{R}^{n \times m}$ to denote the normalized version of $\mathbf{X}$ with $\mathbf{R}_{ij} = g(\mathbf{X}_{ij})$ and $g(x) = \frac{1}{1+\exp^{-x}}$. A user can upload images to LBSNs. $\mathcal{P}_{u_i}$ denotes the set of images uploaded by u_i. A user can also choose to add locations to images. $\mathcal{P}_{l_j}$ denotes the set of images that are tagged l_j. For example, in Figure 1, $\mathcal{P}_{u_1} = \{p_1, p_2, p_3\}$ and $\mathcal{P}_{l_1} = \{p_5\}$. Then the problem is for-

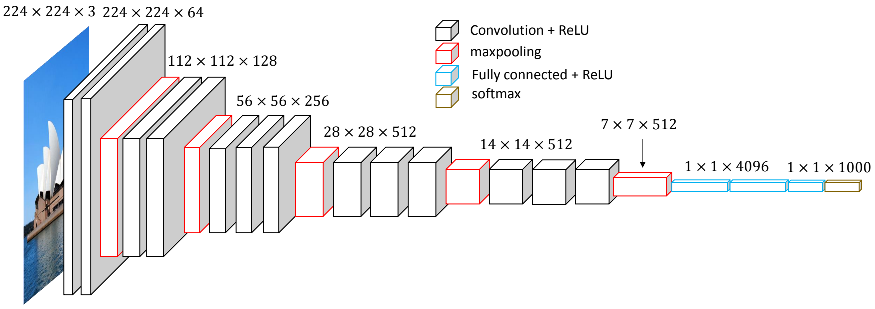


Figure 2: The architecture of VGG16 model .

mally stated as: given check-in matrix $\mathbf{R}$, user images $\mathcal{P}_{u_i}$, $i = 1, \dots, n$ and POI images $\mathcal{P}_{l_j}$, $j = 1, \dots, m$, we aim to recommend k un-visited POIs to each user.

3.1 A Basic POI Recommendation Model

We choose Probabilistic Matrix Factorization (PMF) [25] as the basic model for POI recommendation. PMF is one of the most popular models in collaborative filtering [30, 29] and has been widely adopted for POI recommendation [4, 18]. It assumes Gaussian distribution on the residual noise of observed data as,

$$P(\mathbf{R}|\mathbf{U}, \mathbf{V}, \sigma) = \prod_{i=1}^{n} \prod_{j=1}^{m} [\mathcal{N}(\mathbf{R}_{ij}|\mathbf{u}_i^T \mathbf{v}_j, \sigma^2)]^{\mathbf{Y}_{ij}}, \qquad (1)$$

where $\mathbf{U} \in \mathbb{R}^{K \times n}$ and $\mathbf{V} \in \mathbb{R}^{K \times m}$ are the latent feature matrices of users and POIs, respectively. $\mathcal{N}(x|\mu, \sigma^2)$ is the Gaussian distribution with mean μ and variance σ^2. $\mathbf{Y}$ is the indicator matrix with $\mathbf{Y}_{ij} = 1$ if $\mathbf{R}_{ij} > 0$ and 0 otherwise. PMF also places Gaussian priors on the latent matrices $\mathbf{U}$ and $\mathbf{V}$ as $P(\mathbf{U}|\sigma_u) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{u}_i|0, \sigma_u^2 \mathbf{I})$ and $P(\mathbf{V}|\sigma_v) = \prod_{j=1}^{m} \mathcal{N}(\mathbf{v}_j|0, \sigma_v^2 \mathbf{I})$, where σ_u^2 and σ_v^2 are the variances of the two Gaussian distributions and $\mathbf{I}$ is the identity matrix. Then the posterior distribution can be written as,

$$\begin{aligned} P(\mathbf{U}, \mathbf{V}|\mathbf{R}) = & \prod_{i=1}^{n} \mathcal{N}(\mathbf{u}_i|0, \sigma_u^2 \mathbf{I}) \prod_{j=1}^{m} \mathcal{N}(\mathbf{v}_j|0, \sigma_v^2 \mathbf{I}) \\ & \prod_{i=1}^{n} \prod_{j=1}^{m} [N(\mathbf{R}_{ij}|\mathbf{u}_i^T \mathbf{v}_j, \sigma^2)]^{\mathbf{Y}_{ij}}. \end{aligned} \qquad (2)$$

Note that POI recommendation is one class collaborative filtering, where only positive samples are given. Following the standard way to solve the one-class problem in CF, we sample the same number of unobserved data from the user-poi matrix and treat them as the frequency to 0 [23, 17, 4].

3.2 Extracting and Modeling Visual Contents

To model images for POI recommendation, we first need to extract useful features from images. Convolutional neural network (CNN) is a powerful deep network for extracting high-level visual contents for image classification and object detection. Thus, we choose CNN for feature extraction. We choose VGG16 model as it is the state-of-the-art CNN architecture [26]. Figure 2 gives an illustration of the architecture of VGG16. It is composed of 13 convolution, 5 max pooling, 3 fully connected and 1 softmax layers. The input to VGG16 is an image of size 224x224x3, where 224x224 is the size of the image and 3 is the number of channels, i.e., RGB channels. Thus, we first resize each image to 224x224 as input. Each cubic in the figure is a feature map with the dimension given above it. For example, the leftmost one is the feature map of size 224x224x64 after the connvolution layer. Cubics of the same size has the same dimension. The last layer is softmax layer that is used for classification. We refer readers to [26] for the details of VGG16. We remove the last two layers of VGG16, which are used for classification purpose. Then for an input image p_k, the visual contents are the output of VGG16 with the last two layer removed, which is a vector of dimension $d = 4096$. We denote it as $CNN(p_k)$ because we treat CNN as a feature learning function whose weights will be updated during the learning process. As a common practice, we don't train VGG16 from scratch, instead we use pre-trained VGG16 and then fine-tune certain CNN [24]. More detail will be discussed in Section 4.3. With the image features extracted by CNN, we are going to incorporate these features for POI recommendation.

First, considering an image p_s posted by u_i, it is natural to assume that p_s contains certain visual contents that meets u_i's preferences; while for an arbitrary image p_w posted by other user, i.e., $p_w \notin \mathcal{P}_{u_i}$, p_w is less likely to contain visual contents that meets u_i's preferences. At the same time, u_i's preferences are now captured by the latent features $\mathbf{u}_i$. This implies that $\mathbf{u}_i$ should be able to differentiate if an image p_s is posed by u_i or not based on the visual feature $CNN(p_s)$. With this intuition, we define the probability that p_s belongs to u_i as $P(f_{is} = 1|u_i, p_s)$, where f_{is} denotes if p_s is posted by u_i or not. $P(f_{is} = 1|u_i, p_s)$ is given as

$$P(f_{is} = 1|u_i, p_s) = \frac{\exp(\mathbf{u}_i^T \cdot \mathbf{P} \cdot CNN(p_s))}{\sum_{p_k \in \mathcal{P}} \exp(\mathbf{u}_i^T \cdot \mathbf{P} \cdot CNN(p_k))} \qquad (3)$$

where $\mathbf{P} \in \mathbb{R}^{K \times d}$ is the interaction matrix between the visual contents and latent user features, and d is the dimension of the visual contents. Thus, for $p_s \in \mathcal{P}_{u_i}$, by maximizing $P(f_{is} = 1|u_i, p_s)$, we force $\mathbf{u}_i$ to be similar to the visual contents through the interaction matrix $\mathbf{P}$. In this way, visual contents can guide the learning process of $\mathbf{u}_i$.

Similarly, considering an image p_t associated with l_j, visual contents of p_t is likely to describe POI l_j; while for an arbitrary image p_w not associated with l_j, the visual contents of p_w is less likely to describe l_j. Since l_j is now described by the latent features $\mathbf{v}_j$, $\mathbf{v}_j$ should be able to differentiate if an image p_t is associated with l_j or not based on the visual feature $CNN(p_t)$. Thus, we define the probability that p_t is associated with l_j as $P(g_{jt} = 1|l_j, p_t)$, where g_{jt} denotes if p_t is associated with l_j or not. Similarly, $P(g_{jt} = 1|l_j, p_t)$ is given as

$$P(g_{jt} = 1|l_j, p_t) = \frac{\exp(\mathbf{v}_j^T \cdot \mathbf{Q} \cdot CNN(p_t))}{\sum_{p_k \in \mathcal{P}} \exp(\mathbf{v}_i^T \cdot \mathbf{Q} \cdot CNN(p_k))} \qquad (4)$$

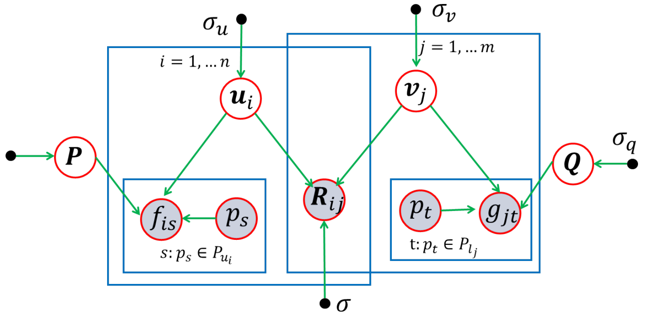


Figure 3: A Graphical Representation of the Model
.

where $\mathbf{Q} \in \mathbb{R}^{K \times d}$ is the interaction matrix between visual contents and latent POI features. Thus, for $p_t \in \mathcal{P}_{l_j}$, by maximizing $P(g_{jt} = 1|l_j, p_t)$, we force $\mathbf{v}_j$ to be close to the visual contents through the interaction matrix $\mathbf{Q}$. In this way, visual contents involve in the learning process of $\mathbf{v}_j$.

With $P(f_{is} = 1|u_i, p_s)$ and $P(g_{jt} = 1|l_j, p_t)$ defined in Eq.(3) and Eq.(4), we can model visual contents by the following likelihood function

$$
\begin{aligned}
&P(\mathcal{F}, \mathcal{G}|\mathcal{P}, \mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}) \\
=&[\prod_{i=1}^{n} \prod_{p_s \in \mathcal{P}_{u_i}} P(f_{is} = 1|u_i, p_s)] \cdot [\prod_{j=1}^{m} \prod_{p_t \in \mathcal{P}_{l_j}} P(g_{jt} = 1|l_j, p_t)]
\end{aligned}
\tag{5}
$$

where $\mathcal{F} = \{f_{is} : p_s \in \mathcal{P}_{u_i} \ \forall \ u_i \in \mathcal{U}\}$ and $\mathcal{G} = \{g_{jt} : p_t \in \mathcal{P}_{l_j} \ \forall \ l_j \in \mathcal{L}\}$. Similarly, we also assume Gaussian priors for $\mathbf{P}$ and $\mathbf{Q}$ as $P(\mathbf{P}|\sigma_p) = \prod_{i=1}^{K} \prod_{j=1}^{d} \mathcal{N}(\mathbf{P}_{ij}|0, \sigma_p^2)$ and $P(\mathbf{Q}|\sigma_q) = \prod_{i=1}^{K} \prod_{j=1}^{d} \mathcal{N}(\mathbf{Q}_{ij}|0, \sigma_q^2)$, where σ_p^2 and σ_q^2 are the variances.

3.3 The Proposed Framework–VPOI

With Eq.(2) modeling user-POI check-in data and Eq.(5) modeling the image features, we propose the visual feature enhanced POI recommendation framework as,

$$
\max_{\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}, CNN} \log P(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}|\mathbf{R}, \mathcal{F}, \mathcal{G}, \mathcal{P}) \tag{6}
$$

where the posterior distribution $P(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}|\mathbf{R}, \mathcal{F}, \mathcal{G}, \mathcal{P})$ can be written as

$$
\begin{aligned}
&P(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}|\mathbf{R}, \mathcal{F}, \mathcal{G}, \mathcal{P}) \\
\propto &P(\mathbf{R}, \mathcal{F}, \mathcal{G}|\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}, \mathcal{P}) P(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}|\mathcal{P}) \\
=&P(\mathbf{R}|\mathbf{U}, \mathbf{V}) P(\mathcal{F}, \mathcal{G}|\mathcal{P}, \mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}) P(\mathbf{P}) P(\mathbf{Q}) P(\mathbf{U}) P(\mathbf{V}).
\end{aligned}
$$

The graphical representation of the model is shown in Figure 3. By substituting Eq.(2), Eq.(5) and equations for priors, the objective function in Eq.(6) can be written as

$$
\begin{aligned}
\max_{\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}, CNN} & - \|\mathbf{Y} \odot (\mathbf{R} - \mathbf{U}^T \mathbf{V})\|_F^2 - \lambda_1 (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \\
& + \alpha \sum_{i=1}^{n} \sum_{p_k \in \mathcal{P}_{u_i}} \log P(f_{ik} = 1|u_i, p_k) - \lambda_2 \|\mathbf{P}\|_F^2 \\
& + \alpha \sum_{j=1}^{m} \sum_{p_k \in \mathcal{P}_{v_j}} \log P(g_{jk} = 1|v_j, p_k) - \lambda_2 \|\mathbf{Q}\|_F^2
\end{aligned}
\tag{7}
$$

where we set $\lambda_1 = \frac{\sigma^2}{\sigma_u^2} = \frac{\sigma^2}{\sigma_v^2}$, $\lambda_2 = \frac{\sigma^2}{\sigma_p^2} = \frac{\sigma^2}{\sigma_q^2}$ to reduce hyperparameters and $\alpha = 2\sigma^2$. $\odot$ is the Hadamard product.

4. AN OPTIMIZATION FRAMEWORK

In this section, we give a framework to solve the optimization problem. We use gradient descent to update the variables alternatively.

4.1 Negative Sampling

The gradients of $\log P(f_{ik} = 1|u_i, p_k)$ and $\log P(g_{jk} = 1|v_j, p_k)$ w.r.t $\mathbf{U}$, $\mathbf{V}$, $\mathbf{P}$, $\mathbf{Q}$ involve the calculation of $\sum_{p_k \in \mathcal{P}} \exp(\mathbf{u}_i^T \cdot \mathbf{P} \cdot CNN(p_k))$ or $\sum_{p_k \in \mathcal{P}} \exp(\mathbf{v}_i^T \cdot \mathbf{Q} \cdot CNN(p_k))$, which requires the summation of all the images and costs a lot of computational operations. To accelerate the speed, following the idea previous work [22, 28], we use negative sampling to approximate $\log P(f_{ik} = 1|u_i, p_k)$ as

$$
\log \sigma(\mathbf{u}_i^T \cdot \mathbf{P} \cdot CNN(p_k)) + \sum_{s=1}^{r} \log \sigma(-\mathbf{u}_i^T \cdot \mathbf{P} \cdot CNN(p_{ks})) \tag{8}
$$

where $p_{ks}, s = 1, \dots, r$, are r negative samples for p_k. p_{ks} are called negative samples. The general idea here is: for each image $p_k \in \mathcal{P}_{u_i}$, we randomly sample r images, i.e., p_{ks}, from images that are not posted by u_i. We then try to maximize the similarity between $\mathbf{u}_i$ and the visual contents of p_k and minimize the similarity between $\mathbf{u}_i$ and p_{ks}. Similarly, $\log P(g_{jk} = 1|v_j, p_k)$ is approximated as

$$
\log \sigma(\mathbf{v}_j^T \cdot \mathbf{Q} \cdot CNN(p_k)) + \sum_{t=1}^{r} \log \sigma(-\mathbf{v}_j^T \cdot \mathbf{Q} \cdot CNN(p_{kt})) \tag{9}
$$

where $p_{kt}, t = 1, \dots, r$, are r images randomly sampled from images not tagged with l_i. With negative sampling, the gradients are simplified, which will be given next.

4.2 Update Rules

To simplify notation, we use $\mathcal{J}$ to denote the objective function in Eq.(7) with the approximations given above.

4.2.1 *Update* U

The partial derivative of $\mathcal{J}$ w.r.t $\mathbf{U}$ is given as

$$
\frac{\partial \mathcal{J}}{\partial \mathbf{U}} = 2\mathbf{V}(\mathbf{Y} \odot \mathbf{R})^T - 2\mathbf{V}[\mathbf{Y} \odot (\mathbf{U}^T \mathbf{V})]^T - 2\lambda_1 \mathbf{U} + \alpha \mathbf{A} \tag{10}
$$

where $\mathbf{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_n\} \in \mathbb{R}^{K \times n}$ is a matrix with its i-th column $\mathbf{a}_i$ given as

$$
\begin{aligned}
\mathbf{a}_i = \sum_{p_k \in \mathcal{P}_{u_i}} \Big[& (1 - \sigma(\mathbf{u}_i^T \mathbf{P} \cdot CNN(p_k))) \mathbf{P} \cdot CNN(p_k) \\
& - \sum_{s=1}^{r} (1 - \sigma(-\mathbf{u}_i^T \mathbf{P} \cdot CNN(p_{ks}))) \mathbf{P} \cdot CNN(p_{ks}) \Big]
\end{aligned}
$$

We can further write $\mathbf{a}_i$ in the vectorized form as

$$
\mathbf{a}_i = \mathbf{P}\mathbf{F}_i(\mathbf{1}_{|\mathcal{P}_{u_i}|} - \sigma(\mathbf{F}_i^T \mathbf{P}^T \mathbf{u}_i)) - \mathbf{P}\tilde{\mathbf{F}}_i(\mathbf{1}_{r \cdot |\mathcal{P}_{u_i}|} - \sigma(-\tilde{\mathbf{F}}_i^T \mathbf{P}^T \mathbf{u}_i))
$$

where $\mathbf{F}_i \in \mathbb{R}^{d \times |\mathcal{P}_{u_i}|}$ is a matrix with each column being $CNN(p_k)$, $p_k \in \mathcal{P}_{u_i}$. Similarly, $\tilde{\mathbf{F}}_i \in \mathbb{R}^{d \times r \cdot |\mathcal{P}_{u_i}|}$ is also a matrix with each column being $CNN(p_{ks})$ where p_{ks} is the negative samples corresponding to $p_k \in \mathcal{P}_{u_i}$. $\mathbf{1}_x$ is an all one vector of length x.

4.2.2 *Update* V

The partial derivative of J w.r.t $\mathbf{V}$ is given as

$$
\frac{\partial \mathcal{J}}{\partial \mathbf{V}} = 2\mathbf{U}(\mathbf{Y} \odot \mathbf{R}) - 2\mathbf{U}[\mathbf{Y} \odot (\mathbf{U}^T \mathbf{V})] - 2\lambda_1 \mathbf{V} + \alpha \mathbf{B} \tag{11}
$$

where $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\} \in \mathbb{R}^{k\times m}$ is a matrix with its j-th column $\mathbf{b}_j$ given as

$$\mathbf{b}_j = \mathbf{Q}\mathbf{K}_j(\mathbf{1}_{|\mathcal{P}_{l_j}|} - \sigma(\mathbf{K}_j^T\mathbf{Q}^T\mathbf{v}_j)) - \mathbf{Q}\tilde{\mathbf{K}}_i(\mathbf{1}_{r\cdot|\mathcal{P}_{l_j}|} - \sigma(-\tilde{\mathbf{K}}_j^T\mathbf{Q}^T\mathbf{v}_j))$$

and $\mathbf{K}_j \in \mathbb{R}^{d\times|\mathcal{P}_{l_j}|}$ is a matrix with each column being $CNN(p_k)$, $p_k \in \mathcal{P}_{l_j}$. Similarly, $\tilde{\mathbf{K}}_i \in \mathbb{R}^{d\times r\cdot|\mathcal{P}_{l_j}|}$ is also a matrix with each column being $CNN(p_{kt})$ where p_{ks} is the negative samples corresponding to $p_k \in \mathcal{P}_{l_j}$.

4.2.3 Update $\mathbf{P}$ *and* $\mathbf{Q}$

The gradient of Eq.(7) w.r.t $\mathbf{P}$ is given as

$$\frac{\partial\mathcal{J}}{\partial\mathbf{P}} = \alpha\sum_{i=1}^{n}\sum_{p_k\in\mathcal{P}_{u_i}}\Big[(1-\sigma(\mathbf{u}_i^T\cdot\mathbf{P}\cdot CNN(p_k)))\mathbf{u}_i CNN(p_k)^T$$
$$-\sum_{s=1}^{r}(1-\sigma(-\mathbf{u}_i^T\cdot\mathbf{P}\cdot CNN(p_{ks})))\mathbf{u}_i CNN(p_{ks})^T\Big] - 2\lambda_2\mathbf{P}$$

which can be written as

$$\begin{aligned}\frac{\partial\mathcal{J}}{\partial\mathbf{P}} &= \alpha\sum_{i=1}^{n}\mathbf{u}_i(\mathbf{1}^T - \sigma(\mathbf{u}_i^T\mathbf{P}\mathbf{F}_i))\mathbf{F}_i^T - 2\lambda_2\mathbf{P}\\ &- \alpha\sum_{i=1}^{n}\mathbf{u}_i(\mathbf{1}^T - \sigma(-\mathbf{u}_i^T\mathbf{P}\tilde{\mathbf{F}}_i))\tilde{\mathbf{F}}_i^T\end{aligned} \tag{12}$$

Similarly, the gradient of Eq.(7) w.r.t $\mathbf{Q}$ is given as

$$\begin{aligned}\frac{\partial\mathcal{J}}{\partial\mathbf{Q}} &= \alpha\sum_{j=1}^{m}\mathbf{v}_j(\mathbf{1}^T - \sigma(\mathbf{v}_j^T\mathbf{Q}\mathbf{K}_j))\mathbf{K}_j^T - 2\lambda_2\mathbf{Q}\\ &- \alpha\sum_{j=1}^{m}\mathbf{v}_j(\mathbf{1}^T - \sigma(-\mathbf{v}_j^T\mathbf{Q}\tilde{\mathbf{K}}_j))\tilde{\mathbf{K}}_j^T\end{aligned} \tag{13}$$

4.2.4 Fine-tune CNN

To update the parameters for CNN, we fix $\mathbf{U}$, $\mathbf{V}$, $\mathbf{P}$ and $\mathbf{Q}$ and remove terms irrelevant to CNN, then the partial derivative of $\mathcal{J}$ w.r.t θ is given as

$$\begin{aligned}\frac{\partial\mathcal{J}}{\partial\theta} =&\sum_{i=1}^{n}\sum_{p_k\in\mathcal{P}_{u_i}}\Big[\Big(1-\sigma\big(\mathbf{u}_i^T\cdot\mathbf{P}\cdot CNN(p_k)\big)\Big)\sum_{h=1}^{d}\mathbf{p}_h^T\mathbf{u}_i\frac{\partial CNN(p_k)_h}{\partial\theta}\\ &-\sum_{s=1}^{r}\Big(1-\sigma\big(-\mathbf{u}_i^T\cdot\mathbf{P}\cdot CNN(p_{ks})\big)\Big)\sum_{h=1}^{d}\mathbf{p}_h^T\mathbf{u}_i\frac{\partial CNN(p_{ks})_h}{\partial\theta}\Big]\\ &+\sum_{j=1}^{m}\sum_{p_k\in\mathcal{P}_{l_j}}\Big[\Big(1-\sigma\big(\mathbf{v}_l^T\cdot\mathbf{Q}\cdot CNN(p_k)\big)\Big)\sum_{h=1}^{d}\mathbf{q}_h^T\mathbf{v}_l\frac{\partial CNN(p_k)_h}{\partial\theta}\\ &-\sum_{t=1}^{r}\Big(1-\sigma\big(-\mathbf{v}_i^T\cdot\mathbf{Q}\cdot CNN(p_{kt})\big)\Big)\sum_{h=1}^{d}\mathbf{q}_h^T\mathbf{v}_l\frac{\partial CNN(p_kt)_h}{\partial\theta}\Big]\end{aligned} \tag{14}$$

where θ is the set of CNN weights to be tuned, which doesn't include the fixed layers. $CNN(p_k)_h$ denotes the h-th element of $CNN(p_k)$. From Eq.(14), we can see that $\frac{\partial\mathcal{J}}{\partial\theta}$ involves the gradients of CNN, while the calculation of gradients of CNN using backpropagation (BP) can be find in [1] and we omit the detail here.

4.3 The Learning Algorithm of VPOI

With the aforementioned update rules, the algorithm of VPOI is summarized in Algorithm 1. In line 1, we first initialize the weights of VGG16 by the pre-trained weights on ImageNet[1] for image classification. ImageNet is a very large image dataset and contains 14,197,122 images with ground truth. It is demonstrated that by initializing CNN using pretrained weights on ImageNet and then fine-tune CNN, we can save computational costs of training and also be able to train a good CNN [24]. In practice, we keep the earlier layers fixed and only fine-tune the last few layers of VGG16. This is motivated by the observation that the earlier features of a ConvNet contain more generic features (e.g. edge detectors or color blob detectors) that should be useful to many tasks, but later layers of the ConvNet becomes progressively more specific to the details of the original dataset and should be fine-tuned for POI recommendation datasets. In line 2, we randomly initialize $\mathbf{U}, \mathbf{V}, \mathbf{P}$ and $\mathbf{Q}$. From line 3 to line 9, we update the parameters until convergence. Finally, for each user u_i, we sort elements in $\mathbf{u}_i^T\mathbf{V}$ in descending order and recommend top-k un-visited POIs.

Algorithm 1 An Optimization Algorithm of VPOI

```
Require: $\mathbf{R}$, $\mathcal{P}_{u_i}$ for $u_i \in \mathcal{U}$, $\mathcal{P}_{l_j}$ for $l_j \in \mathcal{L}$
Ensure: Top-k POIs for each user
 1: initialize VGG16 by using pretrained weights on ImageNet
 2: initialize $\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{Q}$
 3: repeat
 4:     update $\mathbf{U}$ as $\mathbf{U} \leftarrow \mathbf{U} + \eta\frac{\partial\mathcal{J}}{\partial\mathbf{U}}$
 5:     update $\mathbf{V}$ as $\mathbf{V} \leftarrow \mathbf{V} + \eta\frac{\partial\mathcal{J}}{\partial\mathbf{V}}$
 6:     update $\mathbf{P}$ as $\mathbf{P} \leftarrow \mathbf{P} + \eta\frac{\partial\mathcal{J}}{\partial\mathbf{P}}$
 7:     update $\mathbf{Q}$ as $\mathbf{Q} \leftarrow \mathbf{Q} + \eta\frac{\partial\mathcal{J}}{\partial\mathbf{Q}}$
 8:     fine-tune CNN using backpropagation
 9: until convergence
10: return the top-k POIs based on $\mathbf{U}^T\mathbf{V}$
```

4.4 Time Complexity

Let's first consider the time complexity of VGG16. Convolutional and fully connected layers are the most time consuming parts in VGG16, thus, we will focus on time complexity of these two kinds of layers. Let f_l be the number of input channels of the l-th convolutional layer, n_l be the number of filters/channels in the l-th convolutional layer, s_l be the spatial size of the filter and m_l be the spatial size of the output feature map. Then updating filter weights of the l-th convolutional layer for one input costs $\mathcal{O}(f_l s_l^2 n_l m_l^2)$ [12]. Note that for VGG16, the filter size is fixed with 3, i.e., $s_l = 3$. Figure 2, the input image is 224x224x3 and first convolutional feature map (first gray cubic) is 224x224x64, thus we have $f_1 = 3$, $n_1 = 64$, $m_l = 224$ and $s_1 = 3$. Similarly, consider the third convolutional layer (third gray cubic), we have $f_3 = 128$, $n_3 = 56$, $m_l = 112$ and $s_1 = 3$. Therefore, if we fix the first L-convolutional layers, and only fine-tune weights of the last $13 - L$ layers then the cost is $\mathcal{O}((r+1)|\mathcal{P}|\sum_{l=L+1}^{13}(f_l n_l m_l^2))$ in each iteration, where we have neglected $s_l^2 = 9$. We don't consider the cost of the first L convolutional layers as they are fixed don't change during the learning process. The costs of updating weights of the two fully connected layers in one iteration are $\mathcal{O}((r+1)|\mathcal{P}|m_{12}^2 n_{13} d)$ and $(r+1)|\mathcal{P}|\mathcal{O}(d^2)$, respectively. Now let's focus on the time complexity of updating other parameters. Considering the fact that $\mathbf{R}$ is very sparse, the computational cost of $\mathbf{U}$ is mainly the computation of $\mathbf{A}$,

[1] http://www.image-net.org/

which is $\mathcal{O}((r+1)Kd\sum_{i=1}^{n}|\mathcal{P}_{u_i}|)$, where d is the dimension of the visual contents from CNN. Similarly, the cost of updating $\mathbf{V}$ is $\mathcal{O}((r+1)Kd\sum_{j=1}^{m}|\mathcal{P}_{l_j}|)$. The cost of updating $\mathbf{P}$ and $\mathbf{Q}$ are also $\mathcal{O}((r+1)Kd\sum_{i=1}^{n}|\mathcal{P}_{u_i}|)$ and $\mathcal{O}((r+1)Kd\sum_{j=1}^{m}|\mathcal{P}_{l_j}|)$, respectively. Since $\sum_{i=1}^{n}|\mathcal{P}_{u_i}| \leq |\mathcal{P}|$ and $\sum_{j=1}^{m}|\mathcal{P}_{l_j}| \leq |\mathcal{P}|$, the time complexity in each iteration is $\mathcal{O}\big((r+1)\cdot|\mathcal{P}|\cdot[Kd+\sum_{l=L+1}^{13}(f_l n_l m_l^2)+d^2+m_{12}^2 n_{13} d]\big)$. One thing to mention is that usually we only need to tune the weights of last few layers, i.e., $L = 10$ or $L = 11$, and the feature map in the last few layers are not large. Thus, $Kd + \sum_{l=L+1}^{13}(f_l n_l m_l^2) + d^2 + m_{12}^2 n_{13} d$ is also not large significantly.

5. EXPERIMENT

In this section, we conduct experiments to evaluate the effectiveness of the proposed framework VPOI. Specifically, we aim to answer the following three questions:

- Can the proposed framework VPOI improve POI recommendation performance by incorporating images?
- Is VPOI able to mitigate the cold-start problem for POI recommendation by incorporating images? and
- Are visual contents extracted by CNN effective for VPOI compared to other visual contents such as SIFT?

We begin by introducing the datasets and experimental settings, then we compare VPOI with the state-of-the-art POI recommendation systems to answer the first question and we investigate the capability of VPOI in handling the cold-start problem to answer the second question. We then use different visual contents for VPOI to find out the effects of visual contents. Finally, further experiments are conducted to investigate the sensitivity of VPOI to the parameters.

5.1 Datasets and Experimental Settings

We collected two experimental datasets[2], i.e. New York City (NYC) and Chicago (CHI), from a real-world social media site Instagram using Instagram API from Oct. 2015 to Feb. 2016. Instagram allows users to check in at a physical location by posting images and associate the image with geo-tags via his/her cellphone. We crawled the check-ins of users along with the associated images. In addition, we also crawled images posted by users but not explicitly tagged with geo-tags. Following the common way [9], we select check-in locations which have been visited by at least two distinct users, and users who have checked in at least 8 distinct locations. We also remove images that are tagged with "selfie", which we checked manually and find that the majority of images tagged with "selfie" don't contain enough information of POIs or user's interests toward POIs because human body/face takes up almost the whole space of the image. Removing these images can reduce noisy information and improve the performance. The statistics of the final datasets are shown in Table 3. It is evident from the statistics in the table that, both datasets are very sparse, which may cause the data saprsity problem for POI-recommendations; while images are very rich, which have potentials to mitigate the data sparsity and cold-start problems.

Two widely used evaluation metrics, i.e., *precision*@N and *recall*@N are adopted to evaluate the recommendation performance. In our experiment, N is set to 5 and 10, respectively.

For each individual user in the check-in matrix, we randomly select x% of all POIs that he has checked-in for training. The rest of the observed user-POI pairs are used as testing. We also remove the images that are associated with check-ins in the test data to ensure that no information of the test data are exposed during the training process. To investigate the capability of the proposed framework in handling the data sparsity problem, we vary x as $\{20, 40\}$ in this work. The random selection is carried out 10 times independently. The average and standard deviation in terms of *precision*@N and *recall*@N with $N = 5, 10$ are reported.

5.2 Performance Comparison of Recommender Systems

To answer the first question, we compare the proposed system with several representative systems. The comparison results are summarized in Table 1 and Table 2. The representative systems in the table are defined as:

- UCF: User-based collaborative filtering is a state-of-the-art approach for memory-based recommender systems. We adopt the user-based recommender [40] for location recommendation. The interest from u_i to l_j is predicted as an aggregation of check-in frequencies of K most similar users of u_i to p_j. Visual information is not considered.
- VUCF: Visual UCF is a variant of UCF, which uses both visual contents and check-in frequencies to calculate the similarity of two users. Then, the interest from u_i to l_j is predicted as an aggregation of check-in frequencies of K most similar users of u_i to p_j .
- NMF: Non-negative Matrix Factorization [15] is a popular method used for POI recommendation [9, 17]. It decomposes user-POI check-in matrix into two non-negative matrices and predict check-ins with the multiplication of these two matrices.
- PMF: Probabilistic matrix factorization [25] assumes that user preference features and item latent features follow the Guassian distribution. It is our basic location recommendation model,as defined in Eq. (2), without considering the images.
- VBPR: Visual Bayesian personalized ranking [14] incorporates visual contents to Bayesian personalized ranking model. The visual contents are extracted from pre-trained CNN without fine-tuning. These visual contents are directly used as parts of descriptions of POIs to predict preference scores of users w.r.t items. VBPR is not specifically designed for POI recommendation and doesn't consider visual contents for users.

Parameters of all baseline methods are determined via cross validation. For VPOI, we set $\alpha = 0.001$, $K = 10$, $\lambda_1 = \lambda_2 = 1$ and $r = 5$ through the experiments. We use pre-trained VGG16 on ImageNet to initialize the weights[3]. We also fine-tune the last three layers of VGG-16. More details about parameter selection for VPOI will be discussed

[2] The datasets will be publicly available from the first author's homepage

[3] download here: http://www.vlfeat.org/matconvnet/pretrained/

Table 1: Performance comparison on NYC and CHI in terms of Precision@5 and Recall@5.

Dataset	Metric	UCF	VUCF	NMF	PMF	VBPR	VPOI
NYC 20%	prec@5	0.0326±0.0025	0.0377±0.0031	0.0647±0.0034	0.0664±0.0009	0.0697±0.0013	**0.0773**±0.0011
	recall@5	0.0197±0.0015	0.0251±0.0023	0.0391±0.0021	0.0401±0.0006	0.0421±0.0010	**0.0467**±0.0006
NYC 40%	prec@5	0.0440±0.0012	0.0473±0.0021	0.0512±0.0036	0.0509±0.0008	0.0547±0.0018	**0.0618**±0.0008
	recall@5	0.0355±0.0010	0.0385±0.0018	0.0414±0.0029	0.0411±0.0006	0.0430±0.0015	**0.0499**±0.0006
CHI 20%	prec@5	0.043±0.0010	0.0502±0.0015	0.0925±0.0045	0.0911±0.0020	0.1045±0.0013	**0.1126**±0.0010
	recall@5	0.0172±0.0004	0.0235±0.0014	0.0369±0.0018	0.0364±0.0055	0.0418±0.0005	**0.0450**±0.0004
CHI 40%	prec@5	0.0609±0.0050	0.0649±0.0034	0.0926±0.0023	0.0949±0.0044	0.0995±0.0016	**0.1052**±0.0014
	recall@5	0.0324±0.0028	0.0359±0.0025	0.0493±0.0012	0.0505±0.0023	0.0529±0.0009	**0.0560**±0.0008

Table 2: Performance comparison on NYC and CHI in terms of Precision@10 and Recall@10.

Dataset	Metric	UCF	VUCF	NMF	PMF	VBPR	VPOI
NYC 20%	prec@10	0.0288±0.0011	0.0348±0.0014	0.0538±0.0016	0.0558±0.0003	0.0571±0.0013	**0.0606**±0.0005
	recall@10	0.0348±0.0013	0.0405±0.0017	0.065±0.0020	0.0675±0.0004	0.0690±0.0016	**0.0732**±0.0006
NYC 40%	prec@10	0.0318±0.0008	0.0367±0.0015	0.0435±0.0027	0.0422±0.0004	0.0439±0.0009	**0.0472**±0.0005
	recall@10	0.0515±0.0012	0.0562±0.0013	0.0703±0.0043	0.0683±0.0006	0.0710±0.0012	**0.0763**±0.0007
CHI 20%	prec@10	0.0340±0.0005	0.0403±0.00011	0.0789±0.0026	0.0773±0.0090	0.0845±0.0012	**0.0923**±0.0011
	recall@10	0.0272±0.0005	0.0332±0.0015	0.0631±0.0021	0.0618±0.0072	0.0675±0.0008	**0.0738**±0.0009
CHI 40%	prec@10	0.0424±0.0028	0.0475±0.0019	0.074±0.0013	0.0727±0.0048	0.0773±0.0005	**0.0821**±0.0008
	recall@10	0.0451±0.0031	0.0502±0.0024	0.0787±0.0014	0.0774±0.0051	0.0823±0.0007	**0.0874**±0.0009

Table 3: Statistics of the Datasets.

Dataset	NYC	CHI
No. of users	9,893	9,062
No. of POIs	17,153	18,414
No. of check-ins	119,905	159,335
No. of images	464,358	538,830
Check-in Density	7.07×10^{-4}	9.55×10^{-4}

in the following subsections. From the results in the Table 1 and 2, we make the following observations:

- In general, matrix factorization based POI recommendation systems outperform the user-oriented CF method and this observation is consistent with that in [10].

- The proposed framework VPOI obtains better performance than baseline methods based on matrix factorization. We perform t-test on these results, which suggests that the improvement is significant. These results indicate that incorporating visual contents can improve the recommendation performance.

- Both UCF and VUCF are user-based collaborative filtering, however, by considering visual contents for finding K similar users, VUCF significantly outperforms UCF. This is because the rating frequency matrix is very sparse, and visual contents can provide complementary information to alleviate the data sparsity problem. This further demonstrates the effectiveness of considering visual contents for POI recommendation.

- Though both VBPR and VPOI utilize visual contents for recommendation, VPOI outperforms VBPR. The differences between VBPR and VPOI include: (i) VBPR only models images for POIs while VPOI considers images for both users and POIs; (ii) Images from Instagram are noisy, which contains a lot of irrelevant information w.r.t POIs. VBPR directly uses them as descriptions of locations to predict preference scores while VPOI uses them to help learn latent features of users and POIs, which is more robust to noise as it doesn't directly influence the preference scores; (iii) VBPR doesn't fine-tune pre-trained CNN that is trained for image classification, while VPOI fine-tunes CNN to make the extracted features adoptive to POI recommendation.

Via aforementioned analysis, we can draw an answer to the first question – our framework VPOI can significantly improve POI recommendation performance via incorporating visual contents.

5.3 Capability of Handling Cold-Start Users

To answer the second question, we investigate the capability of the proposed framework VPOI in handling cold-start users. *Note that for POI recommendation, a cold-start user means a user who doesn't have check-in history.* Thus, a user who only post photos without adding geo-tag are also considered as cold-start user as we lack check-in history of this user. It is reported that less than 30% images are explicitly tagged with POIs in Instagram[16], which is also consistent with observations from our datasets. Thus, the ability of a recommender system making recommendation for cold-start users are important and necessary. In detail, for each individual user, we first randomly select x% of all POIs that he has checked-in for training. The rest of the observed user-POI pairs are used as testing data. We then remove the images that are associated with check-ins in the test data. After that, we randomly select 5% users from the training set and remove their check-ins from the training set. We also remove images explicitly tagged with geo-locations that are posted by these 5% users. *In this way, these 5% users don't have any check-in history and don't have any images that are explicitly tagged with geo-locations; thus we consider these users as cold-start users.* These 5% users still have images that aren't explicitly tagged with POIs, which can help to reveal their interests and have potentials to mitigate the cold-start problem. For those baseline methods that cannot handle cold-start users, we randomly guess their check-ins for cold-start users. The results with cold-start users are summarized in Table 4 and Table 5. From the tables, we make the following observations:

Table 4: Performance comparison on NYC and CHI with 5% cold-start users in terms of Precision@5 and Recall@5. Note that numbers inside parentheses in the table denote the performance reductions compared to the perforamnce without cold-start users in Table 1.

Dataset	Metric	UCF	VUCF	NMF	PMF	VBPR	VPOI
NYC 20%	prec@5	0.0304(6.75%)	0.0358(5.03%)	0.0606(6.34%)	0.0623(6.17%)	0.0662(5.02%)	**0.0754**(2.46%)
	recall@5	0.0184(6.60%)	0.0239(4.78%)	0.0366(6.39%)	0.0377(5.99%)	0.0400(4.99%)	**0.0456**(2.36%)
NYC 40%	prec@5	0.0419(4.77%)	0.0453(4.23%)	0.0485(5.27%)	0.0475(6.68%)	0.0511(6.58%)	**0.0597**(3.40%)
	recall@5	0.0339(4.51%)	0.0369(4.16%)	0.0392(5.31%)	0.0384(6.57%)	0.0413(3.95%)	**0.0483**(3.21%)
CHI 20%	prec@5	0.0362(15.81%)	0.0457(8.96%)	0.0865(6.49%)	0.0859(5.71%)	0.0975(6.70%)	**0.1098**(2.49%)
	recall@5	0.0144(16.28%)	0.0215(8.51%)	0.0345(6.50%)	0.0343(5.77%)	0.0390(6.70%)	**0.0439**(2.44%)
CHI 40%	prec@5	0.0582(4.43%)	0.0622(4.16%)	0.0887(4.21%)	0.0912(3.90%)	0.0948(4.72%)	**0.1015**(3.52%)
	recall@5	0.0310(4.32%)	0.0344(4.18%)	0.0472(4.26%)	0.0485(3.96%)	0.0504(4.73%)	**0.0540**(3.57%)

Table 5: Performance comparison on NYC and CHI with 5% cold-start users in terms of Precision@10 and Recall@10. Note that numbers inside parentheses in the table denote the performance reductions compared to the performance without cold-start users in Table 2.

Dataset	Metric	UCF	VUCF	NMF	PMF	VBPR	VPOI
NYC 20%	prec@10	0.0264(8.33%)	0.0324(6.89%)	0.0506(5.95%)	0.0529(5.20%)	0.0542(5.08%)	**0.0589**(2.81%)
	recall@10	0.0318(8.62%)	0.0376(7.16%)	0.0612(5.85%)	0.0639(5.33%)	0.0655(5.07%)	**0.0712**(273%)
NYC 40%	prec@10	0.0301(5.35%)	0.0349(4.90%)	0.041(5.75%)	0.0391(7.35%)	0.0422(3.87%)	**0.0456**(3.39%)
	recall@10	0.0487(5.44%)	0.0534(4.98%)	0.0662(5.83%)	0.0632(7.47%)	0.0682(3.94%)	**0.0737**(3.41%)
CHI 20%	prec@10	0.0306(10.00%)	0.0367(8.93%)	0.0709(10.14%)	0.0686(11.25%)	0.0817(3.31%)	**0.0898**(2.71%)
	recall@10	0.0244(10.29%)	0.0301(9.34%)	0.0567(10.14%)	0.0549(11.17%)	0.0654(3.11%)	**0.0718**(2.71%)
CHI 40%	prec@10	0.0404(4.72%)	0.0454(4.42%)	0.0703(5.00%)	0.069(5.09%)	0.0741(4.14%)	**0.0793**(3.41%)
	recall@10	0.0430(4.66%)	0.0480(4.38%)	0.0748(4.96%)	0.0734(5.17%)	0.0789(4.13%)	**0.0844**(3.43%)

- The performance of all methods degenerates when we introduce cold-start users. For example, the performance for PMF decreases up to 11.25% in terms of *precision*@10 on CHI 20%.

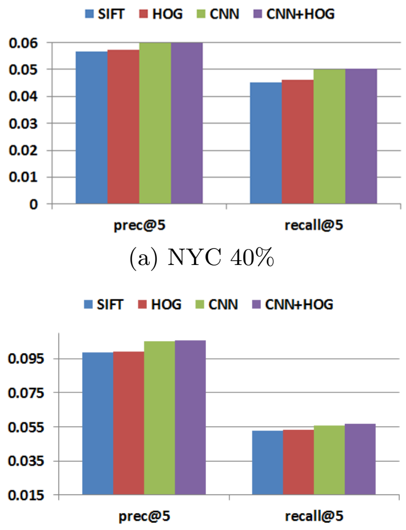
- The performance reduction of VUCF is smaller than UCF. This is because for VUCF, we can use visual contents to find K most similar users for cold-start users; while for UCF, since we are lack of visiting histories of cold-start users, we cannot do prediction for these users.

- Compared to the other methods, the performance reduction of VBPR and the proposed framework VPOI are much smaller and the performance degeneration of VPOI is smaller than VBPR. As aforementioned, the proposed framework can learn user latent factors for cold-start users. These results support that the proposed framework can mitigate the cold-start problem for POI recommendations.

In summary, the introduction of cold-start users could degrade POI recommendation performance and the proposed framework is more robust to cold-start users by incorporating visual contents.

5.4 Effects of Different Visual Contents on VPOI

To answer the third question, we conduct POI recommendation using VPOI with other traditional visual contents, i.e., SIFT [20] and HOG [7], which are two popular manually crafted visual descriptors before the emergence of deep CNN features. To get the features, we first re-size the input image to 224x224x3 so that each image has the same size. We then use VLFEAT [4], a visual feature extraction toolbox, to extract SIFT and HOG features for each image. For each kind of visual contents, we fix the parameter of VPOI as $\alpha = 0.001, K = 10, \lambda_1 = \lambda_2 = 1$ and $r = 5$. The POI recommendation perform of VPOI with different visual contents in terms of precision@5 and recall@5 are reported in Figure 4. From the figure, we observe that:



(a) NYC 40%

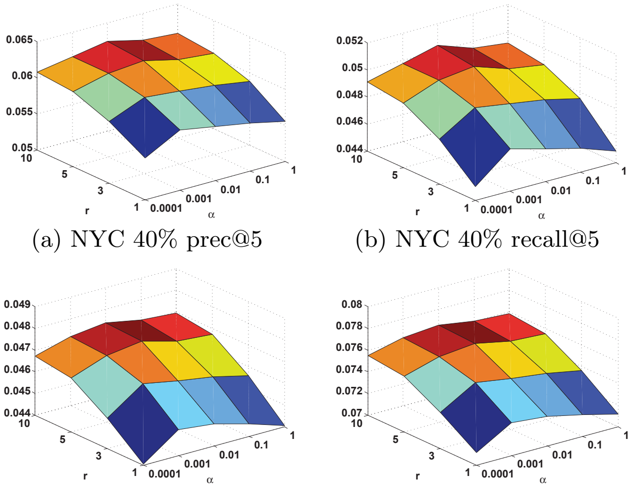
(b) CHI 40%

Figure 4: Performance of VPOI with Different Visual Contents.

- Compared with PMF in Table 4, VPOI with SIFT, HOG or CNN features all outperforms PMF, which implies that visual contents do provide complementary information for POI recommendation.

- CNN visual contents outperforms SIFT and HOG, which is because CNN is pretrained on ImageNet which is able to extract high-level discriminative features, while SIFT and HOG are manually crafted low-level features

[4] http://www.vlfeat.org/



(a) NYC 40% prec@5 (b) NYC 40% recall@5

(c) CHI 40% prec@5 (d) CHI 40% recall@5

Figure 5: Parameter Sensitivity of VPOI w.r.t to α and r.

which are not so discriminative. However, by combing both CNN and HOG features, we notice that the performance improves a little bit, which implies that high level and low level features together can give better result.

5.5 Parameter Sensitivity

The proposed framework has two important parameters, α and r, where α controls the contribution of images in learning the latent features of users and POIs, and r controls the accuracy of Eq.(8) and (9) in approximating Eq.(3) and Eq.(4). In this section, we investigate the impact of the parameters α and r on the performance of VPOI. We only show results on NYC and CHI with 40% without cold-start users since we have similar observations with other experimental settings. We empirically set the latent dimension $K = 10$, the regularization parameters $\lambda_1 = \lambda_2 = 1$. We vary the values of α as $\{0.0001, 0.001, 0.01, 0.1, 1\}$ and r as $\{1, 3, 5, 10\}$. The results are shown in Figure 5. It can be observed from the figure:

- When we set $\alpha = 0$, the proposed framework VPOI boils down to PMF. When we increase α, we incorporate visual contents for learning the latent features of users and POIs. In most cases, the proposed framework VPOI with $\alpha = 0.0001$ and $r = 1$ obtains much better performance than PMF. These results demonstrate the effectiveness of images for POI recommendation.

- Generally, with the increment of α, the performance tends to first increase and then decrease. The performance is relatively stable at certain region, which ease the parameter selection for VPOI in practice.

- As r increases from 1 to 10, the performance increases and then become stable, which is consistent with the observation in [22]. This suggests that a large value of r can achieve better performance; while large r means more computational cost. Thus, there's trade-off between computational cost and recommendation performance.

6. CONCLUSION

In this paper, we investigate visual contents to advance traditional POI recommender systems. To effectively utilizes visual contents, we use CNN to extract features from images and use it to guide the learning process of latent user and POI features, which leads to a novel framework VPOI. Experimental results show that the proposed framework outperforms representative state-of-the-art POI recommender systems. Further experiments are conducted to demonstrate the capability of the proposed framework in mitigating the cold-start problem for recommendation by incorporating images.

There are several directions needing further investigation. First, the proposed VPOI is a flexible framework that is easy to incorporate geographical influence, social correlations, temporal patterns and textual content indications. Thus, we would like to incorporate these factors to see if they can give better performance together with visual contents. For example, social dimensions [27], which captures the affliction of users to different groups, may help to capture the common preferences of users in the same group for POI recommendation. Second, as user check-in records are streaming data, another direction is to extend VPOI using streaming recommender system techniques [2, 3].

7. ACKNOWLEDGEMENTS

This material is based upon work supported by, or in part by, the NSF grants #1614576 and IIS-1217466, and the ONR grant N00014-16-1-2257.

8. REFERENCES

[1] Jake Bouvrie. Notes on convolutional neural networks. 2006.

[2] Shiyu Chang, Yang Zhang, Jiliang Tang, Dawei Yin, Yi Chang, Mark A Hasegawa-Johnson, and Thomas S Huang. Positive-unlabeled learning in streaming networks. In *SIGKDD*, pages 755–764. ACM, 2016.

[3] Shiyu Chang, Yang Zhang, Jiliang Tang, Dawei Yin, Yi Chang, Mark A Hasegawa-Johnson, and Thomas S Huang. Streaming recommender systems. *arXiv preprint arXiv:1607.06182*, 2016.

[4] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the AAAI*, 2012.

[5] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, 2013.

[6] David J Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. Mapping the world's photos. In *WWW*, pages 761–770. ACM, 2009.

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE, 2005.

[8] Huiji Gao and Huan Liu. Mining human mobility in location-based social networks. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 7(2), 2015.

[9] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the Recommender Systems*, pages 93–100. ACM, 2013.

[10] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Content-aware point of interest recommendation on location-based social networks. In *AAAI*, pages 1721–1727. Citeseer, 2015.

[11] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *Proceedings of the CVPR*, pages 1–8. IEEE, 2008.

[12] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *CVPR*, pages 5353–5360, 2015.

[13] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 507–517. International World Wide Web Conferences Steering Committee, 2016.

[14] Ruining He and Julian McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI*, 2016.

[15] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[16] Xutao Li, Tuan-Anh Nguyen Pham, Gao Cong, Quan Yuan, Xiao-Li Li, and Shonali Krishnaswamy. Where you instagram?: Associating your instagram photos with points of interest. In *CIKM*, 2015.

[17] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of SIGKDD*, pages 1043–1051. ACM, 2013.

[18] Bin Liu and Hui Xiong. Point-of-interest recommendation in location based social networks with topic and location awareness. In *SDM*. SIAM, 2013.

[19] Bo Liu, Quan Yuan, Gao Cong, and Dong Xu. Where your photo is taken: Geolocation prediction for social images. *Journal of the Association for Information Science and Technology*, 65(6):1232–1243, 2014.

[20] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[21] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the SIGIR*, pages 43–52. ACM, 2015.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the NIPS*, pages 3111–3119, 2013.

[23] Rong Pan and Martin Scholz. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In *Proceedings of SIGKDD*, pages 667–676. ACM, 2009.

[24] Ali Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the CVPR Workshops*, pages 806–813, 2014.

[25] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *NIPS*, volume 20, pages 1–8, 2011.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] Jiliang Tang, Suhang Wang, Xia Hu, Dawei Yin, Yingzhou Bi, Yi Chang, and Huan Liu. Recommendation with social dimensions. In *AAAI*, pages 251–257, 2016.

[28] Suhang Wang, Jiliang Tang, Charu Aggarwal, and Huan Liu. Linked document embedding for classification. In *CIKM*, pages 115–124. ACM, 2016.

[29] Suhang Wang, Jiliang Tang, and Huan Liu. Toward dual roles of users in recommender systems. In *CIKM*, pages 1651–1660, 2015.

[30] Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. Exploring implicit hierarchical structures for recommender systems. In *IJCAI*, pages 1813–1819, 2015.

[31] Yilin Wang, Suhang Wang, Jiliang Tang, Huan Liu, and Baoxin Li. Unsupervised sentiment analysis for social media images. In *IJCAI*, pages 2378–2379, 2015.

[32] Yilin Wang, Suhang Wang, Jiliang Tang, Huan Liu, and Baoxin Li. Ppp: Joint pointwise and pairwise image label prediction. In *CVPR*, pages 6005–6013, 2016.

[33] Yilin Wang, Suhang Wang, Jiliang Tang, Guojun Qi, Huan Liu, and Baoxin Li. Clare: A joint approach to label classification and tag recommendation. In *AAAI*, 2017.

[34] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 119–128. ACM, 2013.

[35] M Ye, X Liu, and W Lee. Exploring social influence for recommendation-a probabilistic generative approach. In *Proceedings of SIGIR*, pages 325–334, 2012.

[36] Mao Ye, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 458–461. ACM, 2010.

[37] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of SIGIR*, pages 325–334. ACM, 2011.

[38] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 363–372. ACM, 2013.

[39] Jia-Dong Zhang and Chi-Yin Chow. Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 443–452. ACM, 2015.

[40] Dequan Zhou, Bin Wang, Seyyed Mohammadreza Rahimi, and Xin Wang. A study of recommending locations on location-based social network by collaborative filtering. In *Advances in Artificial Intelligence*, pages 255–266. Springer, 2012.